Adding support for IPv6 in IPv4 client/server apps - sin6_flowinfo and sin6_scope_id fields?

I work on implementing IPv6 support for several applications, but I wondered what are these 2 fields for. There are so few questions about this here so I'm not sure I got it right.

- About scope ID (`sin6_scope_id`) - well, **Q1** , **Q2** , **Q3** and **Q4** gave me idea about the scope ID and I think I get it. So, I'll have to add one more config parameter, to make the scope-id configurable. (I decided to add this here, in case that someone is interested in this). Shortly - scope ID is necessary to uniquely determine which is the device, that should handle the traffic - because there may be several interfaces, with the same IP, but with different (interface?) ID. So far, so good.

- But how about the "flow information" (`sin6_flowinfo`)

  - What is it for? I couldn't find anything interesting about that. I read the RFC but it didn't help me at all.

  - Are there some possible values for `sin6_flowinfo` (like - several values, like flags, which mean something), or it's like the `sin6_scope_id` - may be any value, depending on the device, I'm trying to connect to?

  - Should I worry about it at all, or I my just leave it `0` (as in Beej's Guide to Network Programming . And *yes*, I tried that, it works, but I'm not sure if it works only in this case (if it depends on some network configuration), or it will always work, if it's set to `0` ?

  - Or, maybe, I should make it configurable, I mean - add one more config option and let the user defines it's value?

  - `google` -ing "sin6_flowinfo" gives me struct definitions and man pages, nothing useful about this field. Any interesting source? (understandable one..not RFC :D)

EDIT: Well, after @glglgl 's answer and after the hint, that `sin6_flowinfo` may be obsolete, I found some interesting sources: **RFC: IPv6 Flow Label Specification** , **IETF draft: Flow Label as Transport-Layer Nonce** , **Practical guide for solaris** and **wikipedia** . The field is not obsolete (or I couldn't find such source, that confirms this), but it looks like `0` as value is good enough.

c   sockets   network-programming   ipv6

edited Nov 29 '11 at 13:28   asked Nov 24 '11 at 11:57

Kiril Kirov
24.6k   9   79   140

1   I removed the commentary about downvotes - it's a perfectly good question, don't worry about it. – caf Nov 24 '11 at 12:57

1 Answer

The best way to go is to use `getaddrinfo()` .

Pseudo code:

```
struct addrinfo *restrict hints = { .ai_family = AF_UNSPEC, .ai_socktype =
SOCK_STREAM };
struct addrinfo * res, r;
if (0 == getaddrinfo("foo.bar.baz", "http", &hints, &res)) {
    for (r=res; r; r=r->ai_next) {
        sock = socket(r->ai_family, r->ai_socktype, r->ai_protocol);
        connect(sock, r->ai_addr, r->ai_addrlen);
        if error: continue
        break
    }
}
freeaddrinfo(res);
```

This will take the worry about `sin6_scope_id` from you; which is normally `0` , except if you have link-local addresses like `fe80::1234:56ff:fe78:9abc%eth2` . This `eth2` is converted to the correct scope ID.

`sin6_flowinfo` is obsolete (AFAIK) and thus set to 0 in your resulting `struct addrinfo` 'S `ai_addr` .

answered Nov 24 '11 at 13:38

Yep, I know this option and I have added it, but I want to add possibility for manually setting every option. And I stuck on `sin6_flowinfo`. I'll read the link about `sin6_flowinfo` later, but if it's obsolete and I can just leave it `0`, it would be perfect. Thanks once again :) – Kiril Kirov   Nov 24 '11 at 14:37

Well, I couldn't find a relevant source, which says, it's obsolete, but I agree with the `0` value. I found some interesting links, which I'll post in my question, to accept yours, instead of posting my own. Thanks for the help. – Kiril Kirov   Nov 29 '11 at 9:28

1   @KirilKirov You are right: it is quite the opposite of obsolete: They don't know yet exactly what to do with it ;-) – glglgl Nov 29 '11 at 11:03

1   Exactly - the best definition - "they don't know yet exactly what to do with it" :D:D I love it! :D –   Kiril Kirov   Nov 29 '11 at 11:06